Computer as Universal Machine

In today's world, almost everyone has used a computer in some form or another, from their wristwatch, to their car, to their toaster, or to their laptop or ipod or other electronic device.

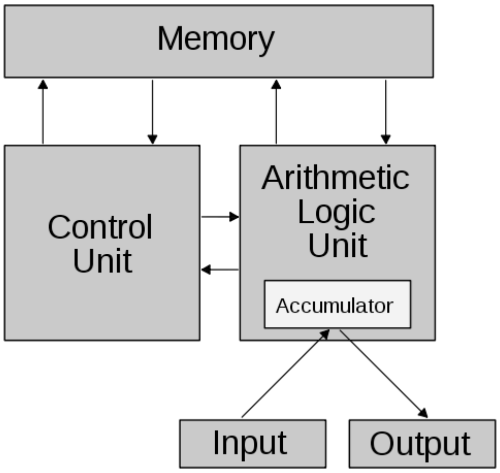
When you think of a computer, what do you think of? That mysterious box that sits on your desk? Your laptop? Your ipod or ipad? Your phone? What do you and others use computers for? Now think about other tools or machines that you use. What do you use these for? Many tools and machines are designed for a specific purpose and it is difficult to use them for anything else. For instance, what is a hammer designed to do? Is it possible to make toast with a hammer? Or vice versa, is it possible to pound nails with a toaster? Well, ok, maybe you could manage to pound a few nails with a toaster, but it wouldn't be long before your toaster was badly damaged or broken.

Computers can take many shapes and sizes. One of the wonderful things about computers is that they can perform many different tasks! A computer is a **universal machine** – one machine that serves many purposes. This is all possible through the use of **programs** to customize the machine for different tasks. The very first computers were machines that were hard-wired to do only one task, so being programmable was a major new step.

A **program** tells a computer exactly what to do through specific instructions. These instructions are very precise – they are in a specific order, they leave nothing out, and they contain no ambiguities. Programs are often built from **algorithms**. An algorithm is a step-by-step solution to some problem. It may be specified in a mathematical notation, or some other general notation. It should give the fundamental ideas necessary for the solution, yet must be detailed enough that it could be translated into a program using a specific language like

Von Neumann Architecture: A major advance

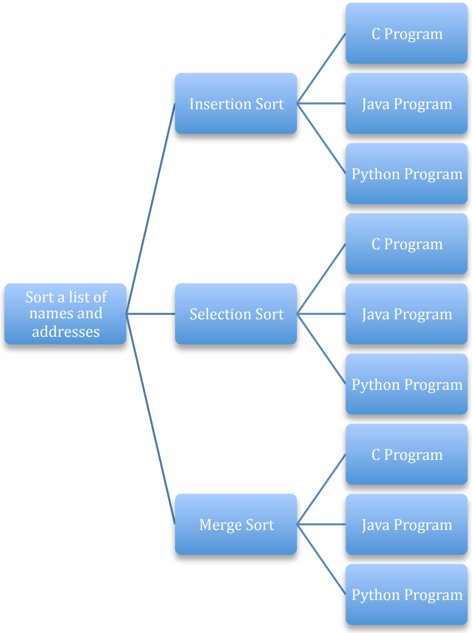
In 1945, mathematician John von Neumann gave the first detailed account of how to build a general purpose computer that would have a single storage structure to hold both instructions and data. The basic design looked like:



The Control Unit and the Arithmetic Logic Unit form the main components of the Central Processing Unit (CPU).

Python, C, Java, Perl, etc. One problem may have several possible algorithmic solutions, each of which may have many translations in different languages.

Example: Suppose that we need to sort a list of names and addresses. Could you come up with a way to do it? Sure, I bet each of us could find a way to accomplish this. If you were to compare your method with your neighbor's, it may not be the same. Sorting is a very important problem in computer science, and there are a number of sorting algorithms that have been shown to work quite well. For each of these algorithms, a programmer could choose any number of languages for the implementation. The diagram below shows us a small example of this.



A **programming language** such as Python, C, Java, or Perl is an English-like notation for specifying a program, using very limited vocabulary and very strict grammar rules. Learning a programming language is very similar to learning a foreign language. As we learn to write programs in Python to manipulate our pictures and sounds, we will need to learn the vocabulary of Python, and its rules for putting words together.

We will learn to write programs in Python, but then how does a computer understand Python or any other language? There must be some type of translator to help the computer understand a particular language. Most languages are either **compiled** or **interpreted** (or some combination of these) into **assembly language**, which is then translated into **machine language**, a language that is directly understood by a computer's hardware systems. If we had to write programs in machine language, we would have to write them with specific codes of 0s and 1s. Different machines would require us to use different codes of 0s and 1s. You can imagine that this would probably not be much fun for very long!